CSE 4510, Section B
Offline 2 on IPC (Inter Process Communication)
Submission Deadline: 12th January 2025, Sunday, 11:59 pm

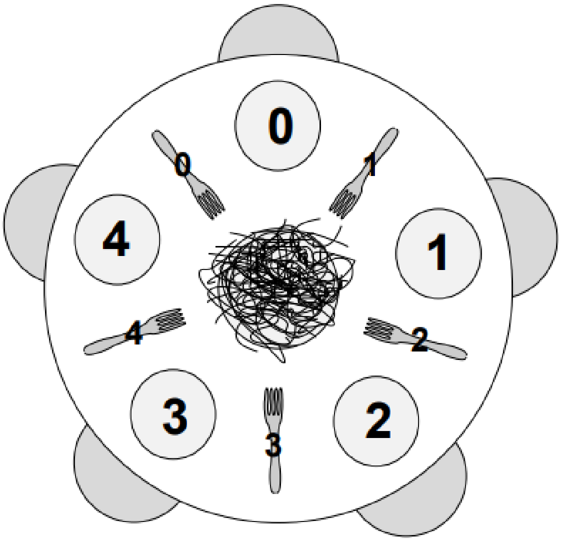
1. Dining Philosophers Problem

The Dining Philosophers Problem was proposed by Dijkstra in 1965, when dinosaurs ruled the earth. It appears in a number of variations, but the standard features are a table with five plates, five forks (or chopsticks) and a big bowl of spaghetti. Five philosophers, who represent interacting threads, come to the table and execute the following loop:

Basic philosopher loop

```
while True:
    think()
    get_forks()
    eat()
    put_forks()
```

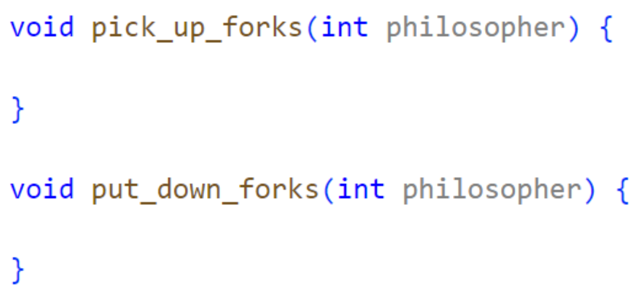
The forks represent resources that the threads have to hold exclusively in order to make progress. Each philosopher alternates between thinking and eating. To eat, a philosopher must use the two forks adjacent to them. After eating, they put down the forks, and then they can start thinking again.



Assuming that the philosophers know how to think and eat, our job is to write a version

of get forks and put forks that satisfies the following constraints:
• Only one philosopher can hold a fork at a time.
• It must be impossible for a deadlock to occur.
• It must be impossible for a philosopher to starve waiting for a fork.
• It must be possible for more than one philosopher to eat at the same time.

The code snippets are given in the dp.c file. You have to modify this file by adding necessary synchronization codes. Understanding the code snippets, you have to



```
void pick_up_forks(int philosopher) {

}

void put_down_forks(int philosopher) {

}
```

implement the following two functions in the dp.c:

2. Implement a synchronization problem where 26 threads each print a letter from A to Z in serial order. This ensures that the letters are printed in alphabetical order from A to Z without any disorder, despite being handled by separate threads.

   Hints:
   1. Use an array of threads (array size will be 26)
   2. Use an array of condition variables (This will make life easier)
   3. The thread that prints the letter A should give signal to a thread that will print letter B.

Protect Your Code:

You should not directly copy the solutions from the Internet, from your friends or any other medium. If copying is found, it will result in -100% marks. That means you will get -10.

Those who have found copying codes earlier, if they are found again to be copying codes, it will result in -200% marks. That means, they will receive -20 marks in this assignment.

Submission Guidelines:

Make a folder using your student id and the folder should contain the following two files:

solution1.c
solution2.c

Zip the folder and submit the zip file in the elms. Do not create any RAR file.

Violation of submission guidelines will result in deduction of 2 marks

Marks Distributions:

Each of the following problems contain 5 marks.